A MODIFIED SEPARABLE PROGRAMMING APPROACH TO WEAPON SYSTEM ALLOCATION PROBLEMS

Thomas Robert McLaughlin

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A MODIFIED SEPARABLE PROGRAMMING
APPROACH TO WEAPON SYSTEM ALLOCATION PROBLEMS

by

Thomas Robert McLaughlin Jr

Thesis Advisor: James G. Taylor

March 1973

Approved for public release; distribution unlimited.

A Modified Separable Programming
Approach to Weapon System Allocation Problems

by

Thomas Robert McLaughlin Jr
Captain, United States Army
B.S., United States Military Academy, 1966

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1973

ABSTRACT

This thesis considers mathematical techniques for computing the optimal allocation of weapons from m different systems against n undefended targets. A standard nonlinear programming problem is considered. A discussion is given on John Danskin's Algorithm for the determination of the optimal values of the lagrange multipliers for this problem. Using a transformation of variables, the nonlinear problem is reformulated as a separable problem and solved by separable programming. A new method, the hybrid algorithm, for the determination of the optimal lagrange multipliers is developed.

TABLE OF CONTENTS

I. INTRODUCTION ---------------------------------- 4

II. MATHEMATICAL MODELS FOR THE ALLOCATION OF WEAPONS TO TARGETS ------------------------ 5

A. AERIAL BOMBING MODEL -------------------- 6

B. "BLACK AND WHITE" TARGET MODEL ----------- 8

C. NUCLEAR WARHEAD MODEL ------------------- 9

III. THE DETERMINATION OF NECESSARY AND SUFFICIENT CONDITIONS ------------------------ 12

IV. NUMERICAL METHODS OF SOLUTION ---------------- 16

A. DANSKIN'S ALGORITHM --------------------- 16

B. SEPARABLE PROGRAMMING ------------------- 18

C. HYBRID ALGORITHM ------------------------ 25

V. EXTENSION OF MODEL AND SOLUTION ------------- 30

APPENDIX A Heuristic Presentation of Danskin's Method with Slight Modification -------- 32

APPENDIX B Worked Numerical Examples of the Separable Programming Method ----------- 44

APPENDIX C Worked Numerical Examples of the Hybrid Method ------------------------- 46

COMPUTER PROGRAM -------------------------------- 51

LIST OF REFERENCES ------------------------------ 59

INITIAL DISTRIBUTION LIST ----------------------- 61

FORM DD 1473 ------------------------------------ 63

I. INTRODUCTION

As a result of today's production costs, armament treaties, and stockpiling facilities, both military tacticians and civilian defense planners are constantly confronted with the decision of how to optimally allocate offensive weapons such as ICBMs, SLBMs, bombs and artillery rounds to various military/industrial targets. Models with varying degrees of complexity and measures of effectiveness have been devised to assist in this decision.

This thesis concerns itself with solving these offensive weapon systems assignment problems so as to optimize measureable returns such as damage or monetary savings. A nonlinear model for an offensive allocation to a group of undefended targets is presented. An algorithm devised by John Danskin [Ref. 1] for obtaining the optimal lagrange multipliers and hence the constrained optima to this problem is discussed and illustrated with an example. The model is then transformed to a separable problem and an approximate solution is obtained utilizing separable programming. Using separable programming and the Kuhn-Tucker conditions, the thesis then proceeds to develop and apply a new method called the hybrid algorithm for calculating the optimal lagrange multipliers and then the optimal allocations.

II. MATHEMATICAL MODELS FOR THE ALLOCATION OF WEAPONS TO TARGETS

The models considered in this paper are those that have a nonlinear objective function and linear constraints. The three specific models presented are those dealing with aerial bombing, "black and white" targets, and nuclear weapons. These three models have two similarities in common. First, the nonlinear function is directly dependent on the military/industrial target value, the number of weapons allocated, and the weapon's effectiveness/ineffectiveness against a target. Secondly, the linear constraint restricts the number of weapons allocated so as not to exceed the total number stockpiled or available.

Similar weapon allocation models, considering some form of cost, have been developed. Examples of these are the allocation of weapons so as to inflict maximum damage with minimum delivery cost, or optimization of production costs subject to a budgetary constraint. This class of models will not be considered in this thesis, but any method presented may be modified to handle such cases.

The basic assumptions relevant to all the following weapon allocation models are:

1. The attack time is of short duration so as to preclude an effectiveness evaluation of preceding rounds.

2. Target location is fixed.

3. Multiple kill by a single round is prohibited.

4. The effectiveness of each individual weapon is independent.

A. AERIAL BOMBING MODEL

This model was developed by Koopman in Ref. 1. It is a model that might be used to allocate weapons, of differing magnitude, delivered by aerial bombardment. The total damage inflicted by the bombardment is related to the number of direct hits by a lethality function. This function reduces the target value by some fractional amount of its previous value. A lethality function representing y direct hits might be written as

$$V(y) = V_j k^y \quad \text{where } 0 \leq k \leq 1 \tag{1}$$

or in terms of damage as

$$D(y) = V_j(1-k^y). \tag{2}$$

If more than one type of weapon is used, the damage function takes on the form

$$D(y) = V_j(1-\prod_{i=1}^{m} k_i^{y_i}). \tag{3}$$

Koopman assumes that each individual weapon (bomb) acts independently and that each bomb of type i has a probability P_{ij} of hitting target j. Thus, the probability that y hits occur out of the x_j bombs dropped on target j is

$$P(y) = \prod_{i=1}^{m} \binom{x_{ij}}{y_i} p_i^{y_i} (1-p_i)^{x_{ij}-y_i} \tag{4}$$

and the expected damage takes the form

$$\bar{d}(x_{ij}) = \sum_{y_i=1}^{x_{ij}} P(y)D(y) = V_j - V_j \left\{ \sum_{y_i=0}^{x_{ij}} \left[\prod_{i=1}^{m} \binom{x_{ij}}{y_i} k_j^{y_i} p_{ij}^{y_i} \left(1-p_{ij}\right)^{x_{ij}-y_i} \right] \right\} . \tag{5}$$

An application of Newton's binomial theorem reduces the expected damage formula to

$$\bar{d}\left(x_{ij}\right) = V_j \left\{ 1 - \prod_{i=1}^{m} \left[1-p_{ij}\left(1-k_j\right) \right]^{x_{ij}} \right\} \tag{6}$$

Thus, the basic model to be used to allocate the aerial weapons is one that maximizes the expected damage subject to the number of bombs available and is stated as

$$\text{maximize} \quad \sum_{j=1}^{n} V_j \left\{ 1 - \prod_{i=1}^{m} \left[1-p_{ij}\left(1-k_j\right) \right]^{x_{ij}} \right\}$$

$$\text{subject to:} \quad \sum x_{ij} \leq X_i \qquad \text{where } i=1,2,\ldots,m \tag{7}$$

$$x_{ij} \geq 0 \qquad \text{and}$$

$$x_{ij} \qquad \text{is an interger.}$$

In order to transform this model into a more mathematically tractable form the following approximation is utilized.

<u>Lemma</u> If N is a large integer and $\left[p(1-k) \right]$ is small, then

$$1-\left[1-p\left(1-k\right)\right]^{N} \approx 1-e^{-\mu N} \qquad \text{where} \quad \mu = p\left(1-k\right)$$

Proof

$$1-\left[1-p\left(1-k\right)\right]^N = 1-\sum_{k=0}^{N}\binom{N}{k}\left(-1\right)^k\left[p\left(1-k\right)\right]^k$$

$$1-\left[1-p\left(1-k\right)\right]^N \approx 1-\left[1-Np\left(1-k\right)\right]$$

and also

$$1-e^{-\mu N} = 1-\sum_{k=0}^{\infty}\left(-1\right)^k\left(\frac{\mu N}{k!}\right)^k \approx p\left(1-k\right)N$$

thus

$$1-e^{-\mu N} \approx 1-\left[1-p\left(1-k\right)\right]^N.$$

Hence, the basic aerial bombing model is now restated as

$$\text{maximize} \quad \sum_{j=1}^{n} V_j\left\{1-\prod_{i=1}^{m} exp\left(-\mu_{ij}\chi_{ij}\right)\right\} \tag{8}$$

$$\text{subject to:} \quad \sum_{j=1}^{n}\chi_{ij} \le X_i \qquad \text{where } i = 1, 2, \ldots, m$$

$$\chi_{ij} \ge 0.$$

B. "BLACK AND WHITE" TARGET MODEL

This model was discussed by Danskin in Ref. 2 and is similar to those published by denBroeder, Ellison, and Emerling [Ref. 3] and Mylander [Ref. 4]. A "black and white" target is one that is eitherdestroyed or not affected by the incoming weapon system, such as missile silos, or other undefended small point targets. Considering the case of m different weapon systems and n "black and white" targets, the kill probability of one weapon from each of these independent weapon systems is

$$1-(1-P_1)\ldots(1-P_m). \tag{9}$$

By letting,

$$1-P_1 = e^{-\mu_1}, \ldots, 1-P_m = e^{-\mu_m} \tag{10}$$

expression (9) is expressed as

$$1-e^{-\mu_1-\ldots-\mu_m} \tag{11}$$

If there were x_i weapons allocated, expression (11) is re-written as

$$1-e^{-\mu_1 x_1-\ldots-\mu_m x_m}, \tag{12}$$

and the basic model for all the "black and white" targets would be stated as

$$\text{maximize } \sum_{j=1}^{n} V_j\left[1-\exp\left(-\sum_{i=1}^{m}\mu_{ij}x_{ij}\right)\right] \tag{13}$$

$$\text{subject to: } \sum_{j=1}^{n} x_{ij} \le x_i \qquad \text{where } i = 1,2,\ldots,m$$

$$x_{ij} \ge 0.$$

C. NUCLEAR WARHEAD MODEL

This model was first discussed and published by Lemus and David [Ref. 5]. It deals with the problem of allocating nuclear weapons of varying yields to relatively defenseless targets. Associated with each weapon system is a launch reliability, R_{ij}, a nuclear yield, y_i, a circular error probable, CEP_i, and a probability of penetrating

defense j, U_{ij}. Given the i^{th} weapon system has been successfully launched and penetrated and j^{th} defense, the target survival probability is

$$Q'_{ij} = \left[\tfrac{1}{2}\right]^{\left(a y_i^{2/3}\right) / H_j^b \left(CEP_i\right)^2} \tag{14}$$

where

a and b are known constants

H_j is the hardness or measure of pressure which a target can resist.

Hence the probability that a target will survive one attacking weapon is

$$Q_{ij} = 1 - P_{ij} = 1 - U_{ij} R_i \left(1 - Q'_{ij}\right) . \tag{15}$$

By assuming a high launch reliability and relatively defenseless target, it follows that

$$R_i U_{ij} \approx 1 \tag{16}$$

and (15) is restated as

$$Q_{ij} = Q'_{ij} = A_j^{P_i} . \tag{17}$$

Thus, the probability that the j^{th} target will survive the attack is given by

$$\prod_{i=1}^{m} Q_{ij}^{x_{ij}} = A_j^{\sum_{i=1}^{m} P_i x_{ij}} = \alpha_{ij}^{\sum_{i=1}^{m} \sigma_i x_{ij}} \tag{18}$$

where $\alpha_{ij} = \left(A_j\right)^{P_1}$ and $\sigma_i = P_i / P_1$

Hence, the basic nuclear weapon model, maximizing the value of the targets destroyed, subject to the restriction that the total weapons allocated does not exceed the number available, is expressed as

$$\text{maximize} \quad \sum_{j=1}^{n} V_j \left[1 - \alpha_{ij}^{\sum_{i=1}^{m} \sigma_i x_{ij}} \right] \tag{19}$$

$$\text{subject to:} \quad \sum_{j=1}^{n} x_{ij} \leq X_i \qquad \text{where } i = 1,2,\ldots,m$$

$$x_{ij} \geq 0.$$

III. THE DETERMINATION OF NECESSARY AND SUFFICIENT CONDITIONS

All subsequent work and calculations, deal only with models of the aerial bombing or "black and white" target form. If the reader desires to use the nuclear warhead model, all formulas and programs must be changed to account for the new constant α_{ij}.

The necessary and sufficient conditions for the optimal solution to the weapon system allocation model, are obtained by a direct application of the Kuhn-Tucker theorem. The reader can find a complete discussion of the Kuhn-Tucker conditions in Refs. 6 and 7. For the model,

$$\text{maximize} \quad \sum_{j=1}^{n} V_j \left[1 - exp\left(-\sum_{i=1}^{m} \mu_{ij} x_{ij} \right) \right]$$

$$\text{subject to:} \sum_{j=1}^{n} x_{ij} \leq X_i \qquad \text{where } i=1,2,,\ldots,m$$

$$x_{ij} \geq 0,$$

the Kuhn-Tucker theorem yields the following conditions

$$x_j^* = 0 \Longleftrightarrow V_j \mu_j e^{-\mu_j x_j^*} - \lambda^* \leq 0 \quad \text{or} \tag{20}$$

$$\mu_j V_j \leq \lambda^*$$

$$x_j^* > 0 \Longleftrightarrow V_j \mu_j e^{-\mu_j x_j} = \lambda^* < \mu_j V_j , \tag{21}$$

thus, $$x_j^* = 1/\mu_j \ln \left(\frac{\mu_j V_j}{\lambda^*}\right) > 0.$$

The following lemma shows, by contradiction, that the lagrange multiplier must be greater than zero and hence the constraint is active.

<u>Lemma</u> The lagrange multiplier, $(\lambda^*) > 0$.

<u>Proof</u> Assume $\lambda^* = 0$.

then $\lambda^* = 0 < \mu_j V_j, \forall_j$,

thus $x_j^* > 0$.

But the Kuhn-Tucker conditions require that

$$x_i^* = \frac{1}{\mu_j} \ln\left[+\infty\right] = +\infty > X \,\text{※}.$$

Hence, λ^* must be greater than zero. The Kuhn-Tucker complementary slackness conditions require that

$$\lambda^* \left(\sum x_j^* - X\right) = 0$$

Since $\lambda^* > 0$, then $\sum_{j=1}^{n} x_j^* = X$. QED

Extending the above to the model consisting of m weapon systems and n targets, the Kuhn-Tucker necessary and sufficient conditions become

$$x_{ij}^* = 0 \Longleftrightarrow \mu_{ij} V_j \exp\left[-\sum_{\substack{k=1 \\ k \neq i}}^{m} \mu_{kj} x_{kj}\right] \leq \lambda_i^* \tag{22}$$

$$x_{ij}^* > 0 \Longleftrightarrow \mu_{ij} V_j \exp\left[-\sum_{i=1}^{m} \mu_{ij} x_{ij}\right] = \lambda_i^*. \tag{23}$$

For any given target j, there are three possible cases that can occur during the allocation process. Either the target will have no weapon system allocated to it, one weapon system allocated, or more than one weapon system allocated. In the first case, for a given target j, all $x^*_{ij}=0$ and the Kuhn-Tucker conditions reduce to the form

$$x^*_{ij}=0 \iff \mu_{ij} V_j \leq \lambda^*_i \qquad \text{for all } i=1,2,\ldots,m \tag{24}$$

The second case requires that $x^*_{kj}=0$ for all weapon systems $k=1,2,\ldots,i-1,i+1,\ldots,m$ and $x^*_{ij}>0$. For this allocation, the Kuhn-Tucker conditions become

$$\begin{aligned} \mu_{ij} V_j \exp\left[-\mu_{ij} x^*_{ij}\right] &= \lambda^*_i \\ \exp\left[-\mu_{ij} x^*_{ij}\right] &= \frac{\lambda^*_i}{\mu_{ij} V_j} \end{aligned} \tag{25}$$

and $$\begin{aligned} \mu_{kj} V_j \exp\left[-\mu_{ij} x^*_{ij}\right] &\leq \lambda^*_k \\ \exp\left[-\mu_{ij} x^*_{ij}\right] &\leq \frac{\lambda^*_k}{\mu_{kj} V_j}\,. \end{aligned} \tag{26}$$

Hence $$\frac{\lambda^*_i}{\mu_{ij} V_j} \leq \frac{\lambda^*_k}{\mu_{kj} V_j} \tag{27}$$

and $$x^*_{ij} = \frac{1}{\mu_{ij}} \ln\left[\frac{\mu_{ij} V_j}{\lambda^*_i}\right]. \tag{28}$$

The final case is just an extension of the preceding one. For this case, the Kuhn-Tucker conditions become

$$\mu_{ij} V_j \exp\left[-\sum_{x_{ij}^*>0} \mu_{ij} x_{ij}^*\right] = \lambda_i^* \qquad \text{for all } i \text{ where } x_{ij}^* > 0 \qquad (29)$$

and $$\mu_{kj} V_j \exp\left[-\sum_{x_{ij}^*>0} \mu_{ij} x_{ij}^*\right] = \lambda_k^* \qquad \text{for all } k \text{ where } x_{kj}^* = 0. \qquad (30)$$

Hence $$\frac{\lambda_i^*}{\mu_{ij}} = \text{a constant} \qquad \text{for all } x_{ij}^* > 0. \qquad (31)$$

Which implies that

$$\frac{\lambda_i^*}{\mu_{ij}} \leq \frac{\lambda_k^*}{\mu_{kj}} \qquad \text{for all } x_{ij}^* > 0,\ x_{kj}^* = 0 \qquad (32)$$

and $$\sum_{x_{ij}^*>0} \mu_{ij} x_{ij}^* = \ln\left[\frac{\mu_{ij} V_j}{\lambda_i^*}\right] \qquad \text{for all } i \text{ where } x_{ij}^* > 0. \qquad (33)$$

IV. NUMERICAL METHODS OF SOLUTION

With the development of high speed computers many analytical and iterative methods have been devised for yielding exact and approximate solutions to the problem. Almost all methods treat the weapons as a continuous variable. Consequently an "eye ball" rounding approximation must be made for the final allocation. This thesis considers the Danskin solution, which is an exact solution, the separable programming approximation, and the hybrid exact solution method.

Other methods available, but not examined include Sequential Unconstrained Minimization Technique (SUMT) by Mylander [Ref. 4], geometeric programming by Passy [Ref.8], treatment as a transportation problem by Manne [Ref. 9], and other analytical algorithms by denBroeder, Ellison, and Emerling [Ref. 3], and Lemus and David [Ref. 5].

A. DANSKIN'S ALGORITHM

One of the earliest algorithms for an exact solution was devised by John Danskin [Ref. 2] in the early 1950's. Danskin's algorithm uses the previously discussed Kuhn-Tucker necessary and sufficient conditions, Gibbs Lemma[1], and the concept of marginal return, to obtain the optimal lagrange multiplier.

---

[1] Gibbs Lemma is discussed in Ref. 2.

The algorithm as applied to the one weapon system n target model consists of the following steps.

Step 1 Consider the quantity $\mu_j V_j$ and arrange the target listing so that $\mu_j V_j \geq \mu_{j+1} V_{j+1}$ for all j.

Step 2 Define the function

$$S(\hat{\lambda}) = \sum_{j=1}^{n} x_j(\hat{\lambda}) = \sum_{\mu_j V_j \geq \hat{\lambda}} \frac{1}{\mu_j} \ln\left[\frac{\mu_j V_j}{\hat{\lambda}}\right], \qquad (34)$$

where $\hat{\lambda}$ denotes a trial value for the optimal value of the lagrange multiplier λ^* and $x_j(\hat{\lambda})$ is a trial allocation. Next find the largest index such that

$$S(\hat{\lambda} = \mu_j V_j) \leq X. \qquad (35)$$

Denote this index as j=L.

Step 3 Once L is known (and hence the targets in the "optimal target list"), λ^* may be explicitly determined. From the Kuhn-Tucker conditions

$$S(\lambda^*) = X = \sum_{j=1}^{L} \frac{1}{\mu_j} \ln\left[\frac{\mu_j V_j}{\lambda^*}\right]$$ and from (36)

step 2

$$S(\hat{\lambda} = \mu_L V_L) = \sum_{j=1}^{L} \frac{1}{\mu_j} \ln\left[\frac{\mu_j V_j}{\mu_L V_L}\right] \qquad (37)$$

both of which contain the same list of regions. Subtracting (37) from (36) we obtain a formula for the optimal lagrange multiplier,

$$\lambda^{*} = \mu_L V_L \exp\left\{ \frac{-[X - S(\mu_L V_L)]}{\sum_{j=1}^{L} \frac{1}{\mu_j}} \right\}. \tag{38}$$

Knowing the index L, λ^{*} may also be calculated by the formula

$$\lambda^{*} = \exp\left\{ \frac{\sum_{j=1}^{L} \frac{1}{\mu_j} \left|\mu_j V_j\right| - X}{\sum_{j=1}^{L} \frac{1}{\mu_j}} \right\}. \tag{39}$$

Step 4 Once λ^{*} has been found, the Kuhn-Tucker necessary and sufficient conditions can be utilized to obtain the optimal allocation.

The algorithm as applied to more than one weapon system is much more complex and requires a computer program to obtain the optimal target listing. A discussion of a typical computer program may be found in Ref. 10. Appendix A, of this thesis gives a heuristic approach to Danskin's algorithm for one and two weapon system model.

B. SEPARABLE PROGRAMMING

Separable programming is used to obtain an approximate solution to nonlinear functions having a separable objective function and constraint. A separable function is any general function that can be written in the form $f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} f_i(x_i)$, where $f_i(x_i)$ is a function of the single variable x_i. A separable problem takes the form

$$\text{maximize} \quad \sum_{i=1}^{n} f_i(x_i) \tag{40}$$

$$\text{subject to:} \quad \sum_{i=1}^{n} g_{ij}(x_i) \leq b_j \qquad \text{for } j=1,2,\ldots,m$$

$$x_i \geq 0 \qquad \text{for all } i.$$

The separable problem is then reduced to a linear programming problem by approximating each separable function by a piecewise linear function. References 6 and 7 should be consulted for a complete development of the piecewise linear approximation.

The separable programming formulation used in this thesis is commonly known as the lambda (λ) method and takes the general form

$$\text{maximize} \quad \sum_{j=1}^{n} \sum_{k=0}^{r_j} \lambda_{kj} f_{kj}(x_j) \tag{41}$$

$$\text{subject to:} \quad \sum_{j=1}^{n} \sum_{k=0}^{r_j} \lambda_{kj} g_{ijk}(x_j) \leq b_i \qquad \text{for } i=1,2,\ldots,m$$

$$\sum_{k=0}^{r_j} \lambda_{kj} = 1 \qquad \text{for } j=1,2,\ldots,n$$

$$\lambda_{kj} \geq 0.$$

The lambda method maximizes (minimizes) the actual piecewise linear function values vice the slope formulation method that maximizes (minimizes) slopes of the approximating function. The slope method was tried for the weapon system allocation problem, but was abandoned because of the time and tedious effort required to update slopes for successive runs.

The weapon system allocation problem (13) can be formulated as a separable programming problem by introducing the new variable $t_j = \sum_{i=1}^{m} \mu_{ij} x_{ij}$, problem (13) can then be restated as[2]

$$\text{maximize} \quad \sum_{j=1}^{n} V_j \left\{ 1 - e^{-t_j} \right\} \tag{42}$$

$$\text{subject to:} \quad t_j - \sum_{i=1}^{m} \mu_{ij} x_{ij} = 0 \qquad \text{for } j=1,2,\ldots,n$$

$$\sum_{j=1}^{n} x_{ij} \le X_i \qquad \text{for } i=1,2,\ldots,m$$

$$x_{ij} \ge 0 \qquad \text{for all } i \text{ and } j,$$

or equivalently as

$$\text{minimize} \quad \sum_{j=1}^{n} V_j e^{-t_j} \tag{43}$$

$$\text{subject to:} \quad t_j - \sum_{i=1}^{m} \mu_{ij} x_{ij} = 0 \qquad \text{for } j=1,2,\ldots,n$$

$$\sum_{j=1}^{n} x_{ij} \le X_i \qquad \text{for } i=1,2,\ldots,m$$

$$x_{ij} \ge 0 \qquad \text{for all } i \text{ and } j.$$

---

2 This approach was suggested by W.M. Raike, Assoc. Professor, Naval Postgraduate School.

By introducing a new variable y_k as

$$
\begin{aligned}
t_1 &\longrightarrow y_1\\
&\vdots\\
t_n &\longrightarrow y_n\\
x_{11} &\longrightarrow y_{n+1}\\
&\vdots\\
x_{1n} &\longrightarrow y_{2n}\\
x_{21} &\longrightarrow y_{2n+1}\\
&\vdots\\
x_{mn} &\longrightarrow y_{(m+1)n},
\end{aligned}
$$

problem (43) is rewritten as

$$\text{minimize} \quad \sum_{k=1}^{n} V_k e^{-y_k} \tag{44}$$

$$\text{subject to:} \quad y_k - \sum_{i=1}^{m} \mu_{ik} y_{ni+k} = 0 \qquad \text{for } k=1,2,\ldots,n$$

$$\sum_{j=1}^{n} y_{ni+j} \le X_i \qquad \text{for } i=1,2,\ldots,m$$

$$y_k \ge 0 \qquad \text{for } k=1,\ldots,(m+1)n.$$

The weapon system allocation problem is now in the separable programming form

$$\text{minimize} \sum_{k=1}^{(m+1)n} f_k(y_k) \tag{45}$$

$$\text{subject to:} \quad \sum_{k=1}^{(m+1)n} g_{jk}\left(y_k\right) = 0 \qquad \text{for } j=1,2,\ldots,n$$

$$\sum_{k=1}^{(m+1)n} g_{jk}\left(y_k\right) \leq X_j \qquad \text{for } j=n+1,n+2,\ldots,n+m$$

$$y_k \geq 0 \qquad \text{for } k=1,2,\ldots,(m+1)n$$

$$\text{where} \quad f_k\left(y_k\right) = \begin{cases} V_k e^{-y_k} & \text{for } k=1,2,\ldots,n \\ 0 & \text{for } k=n+1,n+2,\ldots,(m+1)n \end{cases}$$

for j=1,2,...,n

$$g_{jk}\left(y_k\right) = \begin{cases} y_k & \text{for } k=j \\ -\mu_{ij} y_k & \text{for } k=ni+j \\ & \text{and } i=1,2,\ldots,m \\ 0 & \text{otherwise} \end{cases}$$

and for j=n+1,n+2,...,n+m

$$g_{jk}\left(y_k\right) = \begin{cases} y_k & \text{for } k=n(j-n)+L \\ & \text{and } L=1,2,\ldots,n \\ 0 & \text{otherwise.} \end{cases}$$

Making the final transformation into the lambda form (41), the final problem to solve becomes

$$\text{minimize} \quad \sum_{k=1}^{(m+1)n} \sum_{i=0}^{r_i} \lambda_{ik} f_{ik}\left(y_k\right) \tag{46}$$

$$\text{subject to: } \sum_{k=1}^{(m+1)n} \sum_{i=0}^{r_i} \lambda_{ik} g_{ijk}(y_k) = 0 \quad \text{for } j=1,2,\ldots,n$$

$$\sum_{k=1}^{(m+1)n} \sum_{i=0}^{r_i} \lambda_{ik} g_{ijk}(y_k) \le X_j \quad \text{for } j=n+1,\ldots,n+m$$

$$\sum_{i=0}^{r_i} \lambda_{ik} = 1, \quad \text{for } k=1,2,\ldots,(m+1)n$$

$$\lambda_{ik} \ge 0 \quad \text{for all } i \text{ and } k$$

where

$$f_{ik}(y_k) = \begin{cases} V_k e^{-y_k} & \text{for } k=1,2,\ldots,n \\ 0 & \text{for } k=n+1,n+2,\ldots,(m+1)n \end{cases}$$

for j=1,2,...,n

$$g_{ijk}(y_k) = \begin{cases} y_k & \text{for } k=j \\ -\mu_{ij} y_k & \text{for } k=ni+j \text{ and } i=1,2,\ldots,m \\ 0 & \text{otherwise} \end{cases}$$

and for j=n+1,...,n+m

$$g_{ijk}(y_k) = \begin{cases} y_k & \text{for } k=n(j-n)+L \text{ and } L=1,2,\ldots,n \\ 0 & \text{otherwise.} \end{cases}$$

Since all the separable functions in the objective function of problem (46) are convex, the lambda separable programming method guarantees an approximate optimal solution.

Before discussing the method used to solve this problem, it is necessary to understand the following definitions.

Definition 1 Grid size is number of increments into which the interval representing the range of y_k is subdivided.

Definition 2 Grid refinement is the process of increasing the grid size above that of preceding iterations.

Definition 3 Nesting refinement is the process of reducing the range of the variable y_k about its present solution. The grid size may increase, decrease, or remain constant. For this problem the range of y_k will be plus and minus one previous increment about the present solution.

The optimal solution is obtained by the following lambda (λ) method algorithm.

Step 1 Using the computer program contained herein, generate the approximating function coefficients on punched cards suitable for the IBM MPS/360 program, and a print out of the variable $g_{ijk}(y_k)=y_k$ for $j=n+1,\ldots,n+m$, and $k=n(j-n)+L$ for $L=1,2,\ldots,n$. Instructions for proper input data are contained in the computer program.

Step 2 Place the output cards from step 1 in the IBM MPS/360 program. The λ's associated with the $g_{ijk}(y_k)$ variable described in step 1 begin with column $\left[n \times (\text{Grid size} + 1) + 1 \right]$. Since there is a one to one correspondence between

the λ's and the print out from step 1, the solution for x_{ij} can be found by

$$x_{ij} = g_{ijk}\left(V_k\right) \times \lambda_{ik} \qquad \text{for } j=1,\ldots,n \text{ and } i=1,\ldots,m. \tag{47}$$

Step 3 Repeat steps 1 and 2 using grid or nesting refinement, and continue until a minimum solution is obtained. It should be noted that because of the plateau in the tail of the exponential curve the nesting methods will not work if solutions lie in this area.

In Appendix B, two numerical examples that were solved by this algorithm are given.

C. HYBRID ALGORITHM

Both Danskin's algorithm and the separable programming method have their advantages for the relatively small problems. However, for the larger problems these methods can become lengthy and cumbersome. To help alleviate this problem, the hybrid algorithm was developed. This method utilizes separable programming to obtain a trial target list and the Kuhn-Tucker conditions to calculate trial lagrange multipliers and solution. Refinements are then performed on this trial target list until the Kuhn-Tucker conditions yield optimal lagrange multipliers and solution.

Before proceeding, it is necessary to define four terms peculiar to this algorithm.

Definition 4 A <u>link</u> is a directed arc or branch connecting two weapon systems (nodes) via a shared target.

Definition 5 A <u>beta coefficient</u>,

$$\beta_i = \begin{cases} 1 & \text{for } i=1 \\ \left(\dfrac{\mu_{ij}}{\mu_{nj}}\right) \prod\limits_{\substack{\text{ALL NODES} \\ \text{IN A CHAIN} \\ \text{FROM } i \text{ TO } n}} \beta_k & \text{for } i \neq 1 \text{ and } (n,i) \text{ is link incident to node } i. \end{cases}$$

Definition 6 A <u>Directed Tree</u> is a connected graph which has no circuits or loops. Another definition is that a graph is a tree if and only if every pair of distinct nodes is connected by precisely one path.

Definition 7 A <u>forest</u> is a disconnected graph whose k components are trees.

The hybrid algorithm for solving the weapon system allocation problem, consists of the following steps.

Step 1 Using the lambda separable programming method and a rather coarse grid, generate a trial optimal target listing, i.e. those targets where $x_{ij} > 0$. Make a target by weapon system matrix with the element X denoting $x_{ij} > 0$ and a blank for $x_{ij} = 0$.

Step 2 Draw all possible links between weapon systems, insuring that there are no more than one between any two weapon systems and no weapon system has more than one link incident to it. This procedure will produce a forest of K

directed trees, where each weapon system represents a node. The sum of the weapon systems in all the trees must equal the total available, i.e.

$$\sum_{i=1}^{K} m_K = m \qquad \text{where } m_K \subset m. \quad (48)$$

Step 3 Decompose the forest and consider each tree separately. Designate a weapon system with no links incident to it as the base weapon system in each tree and relabel it number (1). Redesignate the other nodes (weapon systems) in the tree as $2,3,\ldots,m_K$.

Step 4 Calculate the beta coefficient (β_i) for each weapon system in the tree, i.e. $i=1,2,\ldots,m_K$.

Step 5 Calculate the optimal lagrange multiplier for the base weapon system from the following formula;

$$\lambda_i^* = \exp\left\{ \frac{\displaystyle\sum_{i=1}^{m_K} \sum_{\substack{\forall X_{ij}>0 \,\&\\ \text{TGTS NOT}\\ \text{SHARED}}} \beta_i\left(\frac{1}{\mu_{ij}}\right) \ln\left[\frac{\mu_{ij}V_j}{\beta_i}\right] + \sum_{\substack{k\forall \text{SHARED}\\ \text{TGTS, 1}\\ \text{SYSTEM/}\\ \text{TGT}}} \beta_i\left(\frac{1}{\mu_{ik}}\right) \ln\left[\frac{\mu_{ik}V_k}{\beta_i}\right] - \sum_{i=1}^{m_K} \beta_i X_i}{\displaystyle\sum_{i=1}^{m_K} \sum_{\substack{\forall X_{ij}>0 \,\&\\ \text{TGTS NOT}\\ \text{SHARED}}} \beta_i\left(\frac{1}{\mu_{ij}}\right) + \sum_{\substack{k\forall \text{SHARED}\\ \text{TGTS, 1}\\ \text{SYSTEM}\\ \text{TGT}}} \beta_i\left(\frac{1}{\mu_{ik}}\right)} \right\} \quad (49)$$

The above equation was derived by using inductive reasoning on various examples. It should be noted that the second summation in both the

numerator and denominator is taken over all shared targets with the index i representing only one weapon system per shared target.

Step 6 Calculate λ_i^* for all the weapon systems in the tree from the equation

$$\lambda_i^* = \lambda_1^* \beta_i \tag{50}$$

Step 7 Repeat steps 3 through 8 for all the trees in the forest.

Step 8 Using the Kuhn-Tucker conditions solve for all the x_{ij}^*'s by the following formulas.

$$x_{ij}^* = \frac{1}{\mu_{ij}} \ln\left[\frac{\mu_{ij} V_j}{\lambda_i^*}\right] \quad \text{where j is not shared} \tag{51}$$

and

$$\sum_{\substack{\forall \text{SYSTEMS} \\ \text{SHARING} \\ \text{TGT } j}} \mu_{ij} x_{ij}^* = \ln\left[\frac{\mu_{kj} V_j}{\lambda_k^*}\right] \quad \text{where j is shared and k is any system sharing target j.} \tag{52}$$

Step 9 Check to see if all remaining Kuhn-Tucker conditions are satisfied. If satisfied, stop. If not, refine the separable program and repeat steps 1 through 8.

The hybrid method, like the separable programming method, has difficulty handling problems whose solution lies in the tail of the exponential curve. This is caused by the computer's rounding errors and its inability to discriminate between near zero values. The hybrid method in its present configuration is incapable of solving problems

with lower bounds. However, by rederiving the Kuhn-Tucker conditions and equation (49), this method can be modified to handle the additional constraints. Appendix C gives applications of the hybrid method to problems of varying sizes.

V. EXTENSION OF MODEL AND SOLUTION

The models discussed and used in this thesis are not inclusive nor necessarily representative of the present real world situation. Models are constantly changing in order to reflect current technological and strategic advances. A model of current interest is one that considers an allocation of weapons against point and area defenses. For a recent article addressing this problem, refer to Miercort, F.A., Soland, R.M., Optimal Allocation of Missiles Against Area and Point Defenses [Ref. 11]. Readers interested in building new models or modifying existing ones should consult Kooharian, A., Saber, N., and Young, H., A Force Effectiveness Model With Area Defense of Targets [Ref. 12], Perkins, F.M., Optimum Weapon Deployment For Nuclear Attack [Ref. 13], and Day, R.H., Allocating Weapons to Target Complexes By Means of Nonlinear Programming [Ref. 14], Eckler, A. and Burr A., Mathematical Models of Target Coverage and Missiles Allocation [Ref. 15]. For a discussion of defensive models and methods of solution, the reader is referred to Dobbie, J.M., On The Allocation of Effort Among Deterrent Systems [Ref. 16], Brodheim, E., Herzer, I., Russ, L.M., A General Dynamic Model For Air Defense [Ref. 17], and Swenson, G.E., Anti-Ballistic Missile Allocations to Defend Targets With Time Varying Value Structures [Ref. 18].

The reader interested in applying and extending the separable programming technique or the hybrid algorithm to other models, or models with additional constraints is referred to Ref. 19 and 20 for additional information on separable programming.

APPENDIX A

The following two examples present an intuitive approach to Danskin's algorithm. Before proceeding with the examples, it is necessary to understand the concept of marginal return (marginal utility). Marginal return is a measure of the change in the objective function for a given change in the independent variable. Symbolically this is expressed as,

$$\text{Maringal return} = \frac{\Delta F}{\Delta X} \qquad \text{(A-1)}$$

If we let x be or approximate a continuous variable, then marginal return becomes,

$$\text{Marginal return} = F' = \frac{df}{dx} \qquad \text{(A-2)}$$

which represents the slope of the objective function at any given level x. It is assumed that any rational man will allocate his independent variable so as to maximize (minimize) his marginal return. With these concepts in mind, examples 1 and 2 are presented.

Example 1

TARGET	WEAPON SYSTEM EFFECTIVENESS	TARGET VALUE
1	.03	1000
2	.2	100
3	.02	500
4	.2	25
5	.2	10
WEAPONS AVAILABLE	80	

Table 1
Parameter Values

Step 1 Find the marginal return of the objective function with respect to the independent variable (x_j); i.e.

$$\text{Marginal return} = \frac{\partial F(x)}{\partial x_j} = \mu_j V_j \exp\left(-\mu_j x_j\right)$$

For the above data this yields,

$$MR\ (x_1) = -30\exp(-.03x_1)$$

$$MR\ (x_2) = -20\exp(-.2x_2)$$

$$MR\ (x_3) = -10\exp(-.02x_3)$$

$$MR\ (x_4) = -5\exp(-.2x_4)$$

$$MR\ (x_5) = -2\exp(-.2x_5)$$

Step 2 Since targets with the largest marginal return are the most attractive and lucrative, the allocator will fire at those first until his weapons are exhausted. When all the x_j's are zero, target one and two have the highest marginal returns; i.e. -30 and -20 respectively. Thus the weapons should be allocated to target 1 until its marginal return equals that of target 2,

$$-30\exp(-.03x_1) = -20\exp(-.2(0)) = -20$$

$$\exp(-.03x_1) = \frac{20}{30}$$

$$x_1 = 1/.03\ \ln(1.5) = 13.516.$$

Since $x_1 = 13.516 < 80$, more weapons can be allocated.

Now targets 1 and 2 are equally attractive since they both have identical marginal returns, -20. Therefore, continue to fire at targets 1 and 2 until their marginal return equals the third highest, target 3,

$$-30\exp(-.03x_1)=-20\exp(-.2x_2)=-10\exp\left[-.02(0)\right]= -10$$

$$x_1 = 1/.03 \ln(3) = 36.62$$

$$x_2 = 1/.2 \ln(2) = 3.466$$

Since $x_1 + x_2 = 40.086 < 80$, more weapons can be allocated.

Now targets 1,2, and 3 are equally attractive since they all have a marginal return of -10. Therefore, continue to allocate to targets 1, 2, and 3 until their marginal return equals the next largest, target 4,

$$-30\exp(-.03x_1)= -20\exp(-.2x_2)= -10\exp(-.02x_3)= -5\exp\left[-.2(0)\right] = -5.$$

Now $x_1 = 59.725$

$x_2 = 6.932$

$x_3 = 34.657.$

Since $x_1 + x_2 + x_3 = 101.314 > 80$, weapons cannot be allocated to targets 1,2 and 3, until their marginal return is equal to -5. Therefore, return to the point where their marginal return equaled -10 (called index L by Danskin), and find a place where their marginal returns

are equal and only 80 weapons have been allocated. This will occur for a marginal return between -10 and -5.

Step 3 Let z be the optimal marginal return between -10 and -5, and allocate to targets 1,2 and 3 until their marginal return equals -z. Thus,

$$-z = -30\exp\left[-.03\left(x_1' + 36.62\right)\right] = -20\exp\left[-.2\left(x_2' + 3.466\right)\right] = -10\exp\left(-.02x_3'\right)$$

$$\exp\left(-.03x_1'\right) = \exp\left(-.2x_2'\right) = \exp\left(-.02x_3'\right) = z/10 = K$$

and hence $x_1' = {}^{-\ln K}/.03$

$x_2' = {}^{-\ln K}/.2$

$x_3' = {}^{-\ln K}/.02.$

Regardless of the constant K, where $^5/_{10} < K < {}^{10}/_{10}$, used the same proportion of weapons x_1', x_2', and x_3' will be used to obtain any marginal return z. Therefore let K equal .9, and

$x_1' = {}^{.1054}/_{.03} = 3.512$

$x_2' = {}^{.1054}/_{.2} = .5268$

$x_3' = {}^{.1054}/_{.02} = 5.268$

Now find the proportions of weapons used.

Proportion of $x_1' = \frac{3.513}{9.31} = .3774$

Proportion of $x_2' = \frac{.527}{9.31} = .0566$

Proportion of $x_3' = \frac{.527}{9.31} = .566$

There are 80-40.086 = 39.914 weapons remaining to be allocated. Of the unallocated weapons 37.7% belong to x_1', 5.57% belong to x_2', and 56.6% belong to x_3'. Therefore the optimal allocations are

$$x_1 = 36.62 + .377(39.914) = 51.684$$

$$x_2 = 3.4658 + .0567(39.914) = 5.7248$$

$$x_3 = 0 + .567(39.914) = 22.591.$$

Example 2

TARGET	WEAPON SYSTEM EFFECTIVENESS 1	2	TARGET VALUE
1	.03	.005	1000
2	.2	.2	100
3	.02	.06	500
4	.2	2.0	25
5	.2	.2	10
WEAPONS AVAILABLE	80	50	

Table II
Parameter Values

Step 1 Calculate the marginal return of the objective function with respect to the independent variable (x_{ij}). For the above data, this gives

$$MR(x_{11}) = -30\exp\left[-(.03x_{11} + .005x_{21})\right]$$

$$MR(x_{12}) = -20\exp\left[-(.2x_{12} + .2x_{22}\right]$$

$$MR(x_{13}) = -10\exp\left[-(.02x_{13} + .06x_{23})\right]$$

$$MR(x_{14}) = -5\exp\left[-(.2x_{14} + 2x_{24})\right]$$

$$MR(x_{15}) = -2\exp\left[-(.2x_{15} + .2x_{25})\right]$$
$$MR(x_{21}) = -5\exp\left[-(.03x_{11} + .005x_{21})\right]$$
$$MR(x_{22}) = -20\exp\left[-(.2x_{12} + .2x_{22})\right]$$
$$MR(x_{23}) = -30\exp\left[-(.02x_{13} + .06x_{23})\right]$$
$$MR(x_{24}) = -50\exp\left[-(.2x_{14} + 2x_{24})\right]$$
$$MR(x_{25}) = -2\exp\left[-(.2x_{15} + .2x_{25})\right]$$

Step 2 Allocate weapon system 1 to the 5 targets. The solution of example 1 gives the needed solution. Weapons are allocated to targets 1,2, and 3 until their marginal return was -6.365. The allocation was

$$x_{11} = 51.678$$
$$x_{12} = 5.724$$
$$x_{13} = 22.586.$$

Because of this allocation the marginal returns are adjusted as follows,

$$MR(x_{11}) = -6.365\exp(-.005x_{21})$$
$$MR(x_{12}) = -6.365\exp(-.2x_{22})$$
$$MR(x_{13}) = -6.365\exp(-.06x_{23})$$
$$MR(x_{14}) = -5\exp(-2x_{24})$$
$$MR(x_{15}) = -2\exp(-.2x_{25})$$
$$MR(x_{21}) = -1.061\exp(-.005x_{21})$$
$$MR(x_{22}) = -6.365\exp(-.2x_{22})$$
$$MR(x_{23}) = -19.095\exp(-.06x_{23})$$
$$MR(x_{24}) = -50\exp(-2x_{24})$$
$$MR(x_{25}) = -2\exp(-.2x_{25}).$$

Step 3 Progress through the target list allocating x_2 to the targets with the largest marginal return, as previously discussed, until the problem is solved without a shared target or until a shared target is encountered. Hence allocate to target 4 until its marginal return equals the next highest, target 3,

$$-50\exp(-2x_{24}) = -19.095\exp\left[-.06(0)\right] = -19.095,$$

$$x_{24} = .4797.$$

There are 49.5203 weapons of type 2 unallocated, and target 3 has the next highest marginal return. Since target 3 already has weapons of type 1 allocated, there is a possibility of a shared target.

Step 4 Target 3 will have weapons of type 2 allocated to it regardless of whether or not it receives weapons of type 1. Therefore remove target 3 from x_1's target list and check to see if in fact it is shared. Adjusting the marginal returns to reflect the weapons of type 2 already allocated yields

$$MR(x_{11}) = -30\exp(-.03x_{11})$$

$$MR(x_{12}) = -20\exp(-.2x_{12})$$

$$MR(x_{14}) = -1.9095\exp(-.2x_{14})$$

$$MR(x_{15}) = -2\exp(-.2x_{15})$$

$$MR(x_{21}) = -5\exp\left[-(.03x_{11} + .005x_{21})\right]$$

$$MR(x_{22}) = -20\exp\left[-(.2x_{12} + .2x_{22})\right]$$

$$MR(x_{23}) = -30\exp\left[-(.02x_{13} + .06x_{23})\right]$$
$$MR(x_{24}) = -19.095\exp\left[-(.2x_{14})\right]$$
$$MR(x_{25}) = -2\exp\left[-(.2x_{15} + .2x_{25})\right].$$

Allocate weapon system 1 according to the newly calculated marginal returns. As a result weapons will be allocated to targets 1 and 2 until their marginal returns equal -3.514,

$$x_{11} = 71.492$$
$$x_{12} = 8.694.$$

Adjusting the marginal returns to reflect the above allocation yields,

$$MR(x_{11}) = -3.514\exp(-.005x_{21})$$
$$MR(x_{12}) = -3.514\exp(-.2x_{22})$$
$$MR(x_{14}) = -1.9095\exp(-2x_{24})$$
$$MR(x_{15}) = -2\exp(-.2x_{25})$$
$$MR(x_{21}) = -.586\exp(-.005x_{21})$$
$$MR(x_{22}) = -3.514\exp(-2x_{22})$$
$$MR(x_{23}) = -30\exp(-.06x_{23})$$
$$MR(x_{24}) = -19.095\exp(-2x_{24})$$
$$MR(x_{25}) = -.2\exp(-.2x_{25}) .$$

Step 5 Continue the allocation of weapon system 2 to the target list. Recall that target 4 has .4797 weapons previously allocated. Allocate to target 3 until its marginal return equals -19.095.

$$-30\exp(-.06x_{23}) = -19.095$$

$x_{23} = 7.624$ and

$x_{24} + x_{23} < X_2$.

The marginal return for target 3 and weapon system 2 is now $-6.33\exp(-.02x_{13})$. Target 3 is still possibly shared since -6.33 is a larger marginal return than -3.514.

Step 6 Continue allocating weapon system 2 until the marginal return of target 3 and 4 is equal to the next largest, -3.514,

$$-50\exp(-2x_{24}) = -30\exp(-.06x_{23}) = -3.514,$$

thus

$x_{23} = 35.726$

$x_{24} = 1.328$ and

$x_{23} + x_{24} < X_2$.

The marginal return of weapon system 1 against target 3 is now,

$$-10\exp\left[-\left[.02x_{13} + .06(35.726)\right]\right] = -1.176\exp(-.02x_{13}).$$

Since the marginal return $-1.176\exp(-.02x_{13}) <$ $-3.514\exp(-.02x_{13})$, target 3 will not be shared in the optimal allocation. Continue allocating weapon system 2 to the target list. Target 2, which is the next largest marginal return has weapons of type 1 already allocated. Hence there is a possibility of a shared target.

Step 7 Repeat steps 4 through 6 using target 2. Delete target 2 from weapon system 1's target list and

allocate to remaining targets. Adjusting the marginal returns to reflect weapon system 2's current allocation yields,

$$MR(x_{11}) = -30\exp\left[-(.03x_{11} + .005x_{21})\right]$$

$$MR(x_{13}) = -1.176\exp\left[-(.02x_{13} + .06x_{23})\right]$$

$$MR(x_{14}) = -.33514\exp\left[-(.2x_{14} + 2x_{24})\right]$$

$$MR(x_{15}) = -2\exp\left[-(.2x_{15} + .2x_{25})\right]$$

$$MR(x_{21}) = -5\exp\left[-(.03x_{11} + .005x_{21})\right]$$

$$MR(x_{22}) = -20\exp\left[-(.2x_{12} + .2x_{22})\right]$$

$$MR(x_{23}) = -3.514\exp\left[-(.02x_{13} + .06x_{23})\right]$$

$$MR(x_{24}) = -3.514\exp\left[-(.2x_{14} + 2x_{24})\right]$$

$$MR(x_{25}) = -2\exp\left[-(.2x_{15} + .2x_{25})\right] .$$

As a result all 80 weapons can be allocated to target 1 until its marginal return equals $-2.72\exp(-.005x_{21})$.

Step 8 Adjust the marginal returns to reflect this current weapon system 1 allocation,

$$MR(x_{11}) = -2.72\exp\left[-(.03x_{11} + .005x_{21})\right]$$

$$MR(x_{13}) = -1.176\exp\left[-(.02x_{13} + .06x_{23})\right]$$

$$MR(x_{14}) = -.3514\exp\left[-(.2x_{14} + 2x_{24})\right]$$

$$MR(x_{15}) = -2\exp\left[-(.2x_{15} + .2x_{25})\right]$$

$$MR(x_{21}) = -.453\exp\left[-(.03x_{11} + 0\ .005x_{21})\right]$$

$$MR(x_{22}) = -20\exp\left[-(.2x_{12} + .2x_{22})\right]$$

$$MR(x_{23}) = -3.514\exp\left[-(.02x_{13} + .06x_{23})\right]$$

$$MR(x_{24}) = -3.514\exp\left[-(.2x_{14} + 2x_{24})\right]$$

$$MR(x_{25}) = -2\exp\left[-(.2x_{15} + .2x_{25})\right].$$

Allocate weapon system 2 to target 2 until its marginal return equals -3.514. As a result $x_{22} = 3.7$ and $x_{22} + x_{23} + x_{24} < X_2$. Thus, more weapons can be allocated to targets 2,3, and 4 until their marginal return equals the next largest, -2. Weapons can only be allocated until their marginal returns equal -2.886 with

$$x_{22} = 9.66$$

$$x_{23} = 39.4$$

$$x_{24} = 1.32.$$

Step 9 Recalculate weapon system 1's marginal return for this current allocation. As a result

$$MR(x_{12}) = -2.8886\exp(-.2x_{12}).$$

Since marginal return of $-2.886 > -2.72$, target 2 will be shared. Hence, the optimal solution for weapon system 1 is allocation to targets 1

and 2, and weapon system 2 is allocation to targets 2, 3, and 4.

Step 10 The optimal marginal returns (λ_1^* and λ_2^*) can be calculated from equation (49), and the following solution can be obtained.

TARGETS	WEAPON SYSTEM 1	2
1	79.132	0
2	.868	8.992
3	0	39.556
4	0	1.442
5	0	0

Table III
Weapon Allocation Solution

APPENDIX B

The following two examples illustrate solutions by the separable programming method. Both solutions were obtained using the nesting refinement method.

Example 1

TARGETS	WEAPONS SYSTEM EFFECTIVENESS: μ_{ij}	TARGET VALUE
1	.10	1000
2	.02	500
3	.05	200
4	.08	100
5	.01	500
WEAPONS AVAILABLE	50	

Table IV
Value of Parameters

TARGETS	GRID SIZE 25.	GRID SIZE 5	GRID SIZE 5	GRID SIZE 10	EXACT SOLUTION
1	25.999	26.399	26.239	26.239	26.2436
2	15.999	15.6001	16.08	16.08	16.0883
3	6.0	6.399	6.559	6.432	6.4353
4	2.00	1.60001	1.12001	1.248	1.2328
5	0	0	0	0	0

Table V
Weapon Assignment: X_{ij}

Example 2

TARGETS	WEAPON SYSTEM EFFECTIVENESS 1	2	TARGET VALUE
1	.03	.005	1000
2	.2	.2	100
3	.02	.06	500
4	.2	2.0	25
5	.2	.2	10
WEAPONS AVAILABLE	80	50	

Table VI
Value of Parameters

TARGETS	WEAPON SYSTEM 1	2
1	79.14089	0
2	.844646	9.00897
3	0	39.5646
4	0	1.42639
5	0	0

Table VII
Weapon Assignment After Four
Runs Using Grid Size 10

APPENDIX C

The following three examples illustrate solutions obtained by the hybrid algorithm.

Example 1

TARGETS	WEAPON SYSTEM EFFECTIVENESS 1	2	3	TARGET VALUE
1	1	5	3	1
2	2	4	4	1
3	3	3	5	1
4	4	2	4	1
5	5	1	3	1
WEAPONS AVAILABLE	1	1	1	

Table VIII
Value of Parameters

Step 1 Use the separable program with a grid equal to .10, to obtain a target by weapon system matrix. Draw all possible links.

TARGET	WEAPON SYSTEM 1	2	3
1		x	
2		x	x
3			x
4	x		x
5	x		

Step 2 Calculate all beta coefficients.

$$\beta_1 = 1$$

$$\beta_2 = \left(\frac{4}{4}\right)\left(\frac{4}{4}\right) = 1$$

$$\beta_3 = \left(\frac{4}{4}\right) = 1$$

Step 3 Calculate λ_1^*.

$$\lambda_1^* = exp\left\{\frac{\frac{ln(5)}{5} + (1)\frac{ln(5)}{5} + \frac{ln(5)}{5} + \frac{ln(4)}{4} + \frac{ln(4)}{4} - 1 - 1 - 1}{\frac{1}{5} + \frac{1}{5} + \frac{1}{5} + \frac{1}{4} + \frac{1}{4}}\right\}$$

$$\lambda_1^* = .29544$$

Step 4 Calculate all λ_i^*'s,

$$\lambda_2^* = \lambda_3^* = .29544.$$

Step 5 Using the Kuhn-Tucker conditions calculate the optimal allocation, and insure that all the Kuhn-Tucker conditions are satisfied.

TARGET	WEAPON SYSTEM ALLOCATION 1	2	3
1	0	.56514	0
2	0	.43426	.21713
3	0	0	.56514
4	.43426	0	.21713
5	.56514	0	0

Table IX
Solution Matrix

Example 2

TARGETS	WEAPON SYSTEM EFFECTIVENESS 1	2	3	4	TARGET VALUE
1	.01	.075	.015	.1	500
2	.001	.01	.1	.01	900
3	.02	.02	.01	.02	700
4	.015	.013	.03	.02	300
5	.1	.019	.04	.1	600
6	.07	.02	.05	.01	100
WEAPONS AVAILABLE	20	25	50	15	

Table X
Value of Parameters

TARGETS	WEAPON SYSTEM ALLOCATION 1	2	3	4
1	0	0	0	15
2	0	0	29.14	0
3	.373	25	0	0
4	0	0	20.3	0
5	19.627	0	0	0
6	0	0	.48	0
β_i^*	1	1	not shared	not shared

Table XI
Solution Matrix After 3 Separable Nesting Runs Using Grids of 5, 5, and 5 Respectively.

Example 3

TARGET	WEAPON SYSTEM EFFECTIVENESS 1	2	3	4	5	6	TARGET VALUE
1	.4	.5	.1	.27	.33	.41	100
2	1.0	.3	.31	.4	.8	.7	250
3	.8	.2	.51	.7	.3	.6	600
4	.2	.3	.4	.75	.6	.25	200
5	.9	.2	.17	.28	.4	.5	300
6	.5	.1	.15	.2	.3	.4	100
7	.1	.2	.23	.18	.24	.4	450
8	.6	.43	.5	.1	.02	.38	500
9	.6	.1	.4	.3	.2	.6	800
10	.2	.33	.3	.4	.53	.5	900
WEAPONS AVAILABLE	10	20	15	5	10	15	

Table XII
Value of Parameters

TARGET	WEAPON SYSTEM ALLOCATION 1	2	3	4	5	6
1	0	6.1897	0	0	0	0
2	.6601	0	0	0	4.18	0
3	0	0	9.2657	.032	0	0
4	0	0	0	4.9679	0	0
5	4.5381	0	0	0	0	0
6	4.7958	0	0	0	0	0
7	0	0	0	0	0	9.5487
8	0	8.3803	1.8999	0	0	0
9	0	0	3.8239	0	0	5.4512
10	0	5.430	0	0	5.82	0
β_i^*	2.008	1	1.1628	1.596	1.606	1.744

Table XIII
Solution Matrix After Separable
Run Using Grid Size 10.

WEAPON SYSTEM ALLOCATION PROBLEM
COEFFICIENT GENERATOR FOR THE
LAMBDA METHOD SEPARABLE PROGRAM

```
C          THIS PROGRAM IS DESIGNED TO BE USED IN CONJUCTION
C          WITH THE SEPARABLE PROGRAMMING METHOD OR THE
C          HYBRID ALGORITHM AS DESCRIBED IN SECTION III PART
C          B AND C RESPECTIVELY.


C        DEMENSION STATEMENTS


C      THIS PROGRAM HAS BEEN DIMENSIONED FOR A MAXIMUM OF 6
C      WEAPON SYSTEMS, 20 TARGETS, AND A GRID SIZE OF 100.
C      DIMENSION STATEMENTS MUST BE CHANGED FOR LARGER
C      PROBLEMS.  THE DIMENSION OF ANSMAT IS CHANGED AS
C      FOLLOWS:
C        NUMBER OF ROWS = (3*NUMBER OF TARGETS)+(NUMBER OF
C                         TARGETS*NUMBER OF WEAPON SYSTEMS)
C                         +(NUMBER OF WEAPON SYSTEMS + 1)
C        NUMBER OF COLUMNS = GRID SIZE + 1


      DIMENSION VAL(20),EFFECT(6,20),UPPERX(6),UPPERY(20),
     2BLOWY(20),WPNTGT(20),ANSMAT(159,101),UPSOLN(6,20),
     3BOTSOL(6,20),UPOLD(6,20),BLOLDX(6,20),SOLN(6,20)


C        INPUT DATA CARDS


C      INPUT DATA CARDS WILL BE READ INTO THE PROGRAM IN
C      THE FOLLOWING ORDER:
C        CARD SET 1:  M=NUMBER OF WEAPON SYSTEMS
C                     N=NUMBER OF TARGETS
C        CARD SET 2:  VAL(K)=VALUE OF TARGET K
C        CARD SET 3:  EFFECT(J,K)=EFFECTIVENESS OF WEAPON
C                     SYSTEM J AGAINST TARGET K
C        CARD SET 4:  UPPERX(J)=AVAILABLE STOCKPILE OF
C                     WEAPON SYSTEM J
C        CARD SET 5:  WPNTGT(K)=NUMBER OF TOTAL WEAPONS THAT
C                     MUST BE USED AGAINST TARGET K
C        CARD SET 6:  DELTA=GRID SIZE
C        CARD SET 7:  SOLN(J,K)=PREVIOUS ALLOCATION SOLUTION
C        CARD SET 8:  UPOLD(J,K)=UPPER LIMIT ON THE RANGE OF
C                     PREVIOUS SOLUTION
C        CARD SET 9:  BLOLDX(J,K)=LOWER LIMIT ON THE RANGE
C                     OF PREVIOUS SOLUTION
C        CARD SET 10: GRID SIZE OF PREVIOUS RUN

C    READ IN THE NUMBER OF WEAPON SYSTEMS AND TARGETS

      READ (5,1000)M,N
 1000 FORMAT(2I10)
```

```
C     READ IN TARGET VALUES, WEAPON SYSTEM EFFECTIVENESS,
C     AVAILABLE STOCKPILE, AND TOTAL WEAPONS TO BE USED
C     AGAINST TARGET K.
C        NOTE: WPNTGT(K) MUST BE ZERO IF USING THE HYBRID
C              ALGORITHM

      READ (5,1100) (VAL(K),K=1,N)
      READ (5,1100) ((EFFECT(J,K),K=1,N),J=1,M)
      READ (5,1100) (UPPERX(J),J=1,M)
      READ(5,1100) (WPNTGT(K),K=1,N)
 1100 FORMAT(10F7.3)

C     READ IN GRID SIZE TO BE USED

      READ (5,1200)DELTA
 1200 FORMAT(F7.1)

C     READ IN PREVIOUS SOLUTION, AND ITS UPPER AND LOWER
C     RANGES.
C        NOTE: FOR GRID REFINEMENT METHOD USE SOLN=MID-RANGE
C              OF WEAPONS AVAILABLE, UPOLD=UPPER LIMIT ON
C              STOCKPILE, AND BLOLDX=ZERO ON ALL RUNS.
C              FOR THE NESTING REFINEMENT USE THE SAME DATA AS
C              THE GRID REFINEMENT FOR THE FIRST RUN. ON ALL
C              SUCEEDING RUNS USE THE ACTUAL SOLUTION FOR
C              SOLN, AND THE UPPER AND LOWER RANGES OF WEAPON
C              SYSTEM J AGAINST TARGET K FROM THE COMPUTER
C              PRINT OUT.

      READ (5,1100) ((SOLN(J,K),K=1,N),J=1,M)
      READ (5,1100) ((UPOLD(J,K),K=1,N),J=1,M)
      READ (5,1100) ((BLOLDX(J,K),K=1,N),J=1,M)

C     READ IN PREVIOUS GRID SIZE
C        NOTE: WHEN USING THE GRID REFINEMENT METHOD ALWAYS
C              READ IN OLDELT=2. WHEN USING THE NESTING
C              REFINEMENT METHOD READ IN OLDELT=2 FOR THE
C              FIRST RUN AND USE THE GRID SIZE FROM THE
C              PRECEEDING RUN FOR EACH SUCEEDING RUN.

      READ (5,1200) OLDELT



C           PRINT OUT SOME PARAMETERS THAT WERE READ INTO THE
C           PROGRAM AND CALCULATE THE NEW RANGES FOR THE
C           FUNCTION GIJK(Y(K))

C        PRINT OUT THE NUMBER OF TARGETS AND WEAPON SYSTEMS

      WRITE(6,1010)M,N
 1010 FORMAT('1','WEAPON SYSTEMS=',I5,/,1X,'TARGETS=',I5,/)

C        PRINT OUT THE TARGET VALUE

      DO 3 K=1,N
      WRITE(6,1110)K,VAL(K)
 1110 FORMAT(1X,'VALUE OF TARGET',I3,2X,'=',F10.3)
    3 CONTINUE
```

```
C       PRINT OUT WEAPON SYSTEM EFFECTIVENESS AND CALCULATE
C       RANGES FOR GIJK(Y(K))

      DO 5 J=1,M
      DO 4 K=1,N
      Y=(UPCLD(J,K)-BLCLDX(J,K))/OLDELT
      UPSOLN(J,K)=SOLN(J,K)+Y
      BOTSOL(J,K)=SOLN(J,K)-Y
      IF(BOTSOL(J,K).GE.0.0) GO TO 112
      BOTSOL(J,K)=0.0
  112 IF(UPSOLN(J,K).LE.UPPERX(J)) GO TO 11
      UPSOLN(J,K)=UPPERX(J)
   11 WRITE(6,1120)J,K,EFFECT(J,K)
 1120 FORMAT(1X,'EFFECTIVENESS OF WEAPON SYSTEM',I5,3X,
     2'AGAINST TARGET',I5,2X,'=',F8.5)
    4 CONTINUE
    5 CONTINUE


C       PRINT OUT WEAPON SYSTEM STOCKPILE

      DO 6 J=1,M
      WRITE(6,1130)J,UPPERX(J)
 1130 FORMAT(1X,'UPPER LIMIT OF WEAPONS OF TYPE',I5,3X,
     2'AVAILABLE=',F8.3)
    6 CONTINUE


C       PRINT OUT TOTAL WEAPONS TO BE USED AGAINST TARGET K

      DO 7 K=1,N
      WRITE(6,1135)K,WPNTGT(K)
 1135 FORMAT(1X,'LOWER LIMIT OF WEAPONS TO BE USED AGAINST
     2TARGET',1X,I5,1X,'=',F10.5)
    7 CONTINUE


C       PRINT OUT GRID SIZE

      WRITE(6,1140)DELTA
 1140 FORMAT(1X,'NUMBER OF INTERVALS=',F8.3)


C         CALCULATE PARAMETERS FOR DO LOOPS

      KK=N+M
      NN=(M+1)*N
      MM=(M+N+1)+(M+1)*N
      KNM=MM+N
      KKNN=KK+NN
```

```
C          PREPARE THE FIRST SET OF CARDS FOR THE MPS SYSTEM

C        CARDS THAT DEFINE TYPE OF CONSTRAINT
         WRITE(7,1500)
 1500    FORMAT('ROWS',/,1X,'N',2X,'C')
         DO 700 I=1,N
         IF(I.GT.99) GO TO 715
         IF(I.GT.9) GO TO 710
         WRITE(7,1510)I
         GO TO 700
  710    WRITE(7,1520)I
         GO TO 700
  715    WRITE(7,1525)I
  700    CONTINUE
         DO 720 J=1,M
         JM=J+N
         IF(JM.GT.99) GO TO 725
         IF(JM.GT.9) GO TO 730
         WRITE(7,1530)JM
 1530    FORMAT(1X 'L',2X,'R',I1)
         GO TO 720
  730    WRITE(7,1540)JM
 1540    FORMAT(1X,'L',2X,'R',I2)
         GO TO 720
  725    WRITE(7,1545)JM
 1545    FORMAT(1X,'L',2X,'R',I3)
  720    CONTINUE
         DO 740 J=1,NN
         JN=J+KK
         IF(JN.GT.99) GO TO 745
         IF(JN.GT.9) GO TO 750
         WRITE(7,1510)JN
 1510    FORMAT(1X,'E',2X,'R',I1)
         GO TO 740
  750    WRITE(7,1520)JN
 1520    FORMAT(1X,'E',2X,'R',I2)
         GO TO 740
  745    WRITE(7,1525)JN
 1525    FORMAT(1X,'E',2X,'R',I3)
  740    CONTINUE
         DO 780 K=1,N
         JO=KKNN+K
         IF(JO.GT.99) GO TO 775
         IF(JO.GT.9) GO TO 770
         WRITE(7,1546)JO
 1546    FORMAT(1X,'G',2X,'R',I1)
         GO TO 780
  770    WRITE(7,1547)JO
 1547    FORMAT(1X,'G',2X,'R',I2)
         GO TO 780
  775    WRITE(7,1548)JO
 1548    FORMAT(1X,'G',2X,'R',I3)
  780    CONTINUE
         WRITE(7,1550)
 1550    FORMAT('COLUMNS')


C          CALCULATE THE NUMBER OF MESH POINTS

         NDELTA=DELTA+1
```

```
C          START THE MAIN DC LOOP FOR CALCULATING COEFFICIENTS

      DO 200 K=1,NN

C      INITIALIZE THE MATRIX FOR STORING THE COEFFICIENTS
C      BY SETTING IT EQUAL TO ZERO

      DO 10 II=1,NDELTA
      DO 10 LL=1,KNM
      ANSMAT(LL,II)=0.0
   10 CONTINUE


C      CALCULATE THE UPPER LIMIT FOR THE FUNCTION FIK(Y(K))
C      WHERE K=1,2,...,N

      IF(K.GT.N) GO TO 31
      OLDBOT=0.0
      OLDUP=0.0
      DO 20 JJ=1,M
      UPPERY(K)=EFFECT(JJ,K)*UPSOLN(JJ,K)+OLDUP
      BLOWY(K)=EFFECT(JJ,K)*BOTSOL(JJ,K)+OLDBOT
      OLDUP=UPPERY(K)
      OLDBOT=BLOWY(K)
   20 CONTINUE
      RY=(UPPERY(K)-BLOWY(K))/DELTA
      CLOWY=BLOWY(K)


C      CALCULATE THE COEFFICIENTS FOR THE FUNCTION FIK(Y(K))
C      WHERE K=1,2,...,N

      DO 30 MA=1,NDELTA
      ANSMAT(1,MA)=VAL(K)*EXP(-CLOWY)
      CLOWY=CLOWY+RY
   30 CONTINUE
   31 CONTINUE


C      CALCULATE THE COEFFICIENTS FOR THE FUNCTION
C         GIJK(Y(K)) WHERE J=1,2,...,(M+N)

      DO 100 J=1,KK
      IF(J.GT.N) GO TO 70
      IF(J.EQ.K) GO TO 35
      GO TO 38

C    CALCULATE THE COEFFICIENTS FOR THE FUNCTION GIJK(Y(K))
C    WHERE J=1,2,...,N AND K=J

   35 KA=K+1
      DO 40 MB=1,NDELTA
      ANSMAT(KA,MB)=BLOWY(K)
      BLOWY(K)=BLOWY(K)+RY
   40 CONTINUE
      GO TO 100

C    CALCULATE THE COEFFICIENTS FOR THE FUNCTION GIJK(Y(K))
C    WHERE J=1,2,...,N, K=NI+J, AND I=1,2,...,N

   38 DO 60 I=1,M
      KB=(N*I)+J
      IF(K.NE.KB) GO TO 60
      RX=(UPSOLN(I,J)-BOTSOL(I,J))/DELTA
      CLOWX=BOTSOL(I,J)
```

```
      JA=J+1
      DO 50 MC=1,NDELTA
      ANSMAT(JA,MC)=(-EFFECT(I,J))*CLOWX
      CLOWX=CLOWX+RX
   50 CONTINUE
   60 CONTINUE
      GO TO 100
   70 CONTINUE

C     CALCULATE THE COEFFICIENTS FOR THE FUNCTION GIJK(Y(K))
C     WHERE J=1N+1,N+2,...,N+M, J=N(J-N)+1, AND L=1,2,...N;
C     AND THE COEFFICIENTS FOR THE TOTAL WEAPONS PER TARGET
C     CONSTRAINT

      DO 90 L=1,N
      KD=(N*(J-N))+L
      IF(K.NE.KD) GO TO 90
      LL=J-N
      LA=J+1
      LB=MM+L
      RZ=(UPSOLN(LL,L)-BOTSOL(LL,L))/DELTA
      DLOWX=BOTSOL(LL,L)
      DO 80 MD=1,NDELTA
      ANSMAT(LA,MD)=DLOWX
      ANSMAT(LB,MD)=DLOWX
      WRITE(6,1145)DLOWX
 1145 FORMAT(80X,F12.6)
      DLOWX=DLOWX+RZ
   80 CONTINUE
   90 CONTINUE
  100 CONTINUE


C       CALCULATE THE COEFFICIENTS FOR THE SUM OF THE LAMBDAS
C          EQUAL 1 CONSTRAINT

      ME=M+N+1+K
      DO 110 LA=1,NDELTA
      ANSMAT(ME,LA)=1.0
  110 CONTINUE
```

```
C         PREPARE THE SECOND SET OF CARDS FOR THE MPS SYSTEM

C        CARDS FOR THE SEPARABLE PROGRAM COEFFICIENTS
      DO 190 L=1,NDELTA
      DO 185 I=1,KNM
      IF(ANSMAT(I,L).EQ.0.0) GO TO 185
      KA=K-1
      JCOLM=(KA*NDELTA)+L
      JROW=I-1
      IF(JCOLM.GT.999) GO TO 115
      IF(JCOLM.GT.99) GO TO 120
      IF(JCOLM.GT.9) GO TO 125
      IF(JROW.GT.999) GO TO 4008
      IF(JROW.GT.99) GO TO 4006
      IF(JROW.GT.9) GO TO 4004
      IF(I.EQ.1) GO TO 4000
      GO TO 4002
  125 IF(JROW.GT.999) GO TO 4018
      IF(JROW.GT.99) GO TO 4016
      IF(JROW.GT.9) GO TO 4014
      IF(I.EQ.1) GO TO 4010
      GO TO 4012
  120 IF(JROW.GT.999) GO TO 4028
      IF(JROW.GT.99) GO TO 4026
      IF(JROW.GT.9) GO TO 4024
      IF(I.EQ.1) GO TO 4020
      GO TO 4022
  115 IF(JROW.GT.999) GO TO 4038
      IF(JROW.GT.99) GO TO 4036
      IF(JROW.GT.9) GO TO 4034
      IF(I.EQ.1) GO TO 4030
      GO TO 4032
 4000 WRITE(7,5000)JCOLM,ANSMAT(I,L)
 5000 FORMAT(4X,'C',I1,8X,'C',9X,F11.5)
      GO TO 185
 4002 WRITE(7,5002)JCOLM,JROW,ANSMAT(I,L)
 5002 FORMAT(4X,'C',I1,8X,'R',I1,8X,F11.5)
      GO TO 185
 4004 WRITE(7,5004)JCOLM,JROW,ANSMAT(I,L)
 5004 FORMAT(4X,'C',I1,8X,'R',I2,7X,F11.5)
      GO TO 185
 4006 WRITE(7,5006)JCOLM,JROW,ANSMAT(I,L)
 5006 FORMAT(4X,'C',I1,8X,'R',I3,6X,F11.5)
      GO TO 185
 4008 WRITE(7,5008)JCOLM,JROW,ANSMAT(I,L)
 5008 FORMAT(4X,'C',I1,8X,'R',I4,5X,F11.5)
      GO TO 185
 4010 WRITE(7,5010)JCOLM,ANSMAT(I,L)
 5010 FORMAT(4X,'C',I2,7X,'C',9X,F11.5)
      GO TO 185
 4012 WRITE(7,5012)JCOLM,JROW,ANSMAT(I,L)
 5012 FORMAT(4X,'C',I2,7X,'R',I1,8X,F11.5)
      GO TO 185
 4014 WRITE(7,5014)JCOLM,JROW,ANSMAT(I,L)
 5014 FORMAT(4X,'C',I2,7X,'R',I2,7X,F11.5)
      GO TO 185
 4016 WRITE(7,5016)JCOLM,JROW,ANSMAT(I,L)
 5016 FORMAT(4X,'C',I2,7X,'R',I3,6X,F11.5)
      GO TO 185
 4018 WRITE(7,5018)JCOLM,JROW,ANSMAT(I,L)
 5018 FORMAT(4X,'C',I2,7X,'R',I4,5X,F11.5)
      GO TO 185
 4020 WRITE(7,5020)JCOLM,ANSMAT(I,L)
 5020 FORMAT(4X,'C',I3,6X,'C',9X,F11.5)
      GO TO 185
 4022 WRITE(7,5022)JCOLM,JROW,ANSMAT(I,L)
 5022 FORMAT(4X,'C',I3,6X,'R',I1,8X,F11.5)
      GO TO 185
 4024 WRITE(7,5024)JCOLM,JROW,ANSMAT(I,L)
 5024 FORMAT(4X,'C',I3,6X,'R',I2,7X,F11.5)
```

```
      GO TO 185
 4026 WRITE(7,5026)JCOLM,JROW,ANSMAT(I,L)
 5026 FORMAT(4X,'C',I3,6X,'R',I3,6X,F11.5)
      GO TO 185
 4028 WRITE(7,5028)JCCLM,JRCW,ANSMAT(I,L)
 5028 FORMAT(4X,'C',I3,6X,'R',I4,5X,F11.5)
      GO TO 185
 4030 WRITE(7,5030)JCCLM,ANSMAT(I,L)
 5030 FORMAT(4X,'C',I4,5X,'C',9X,F11.5)
      GO TO 185
 4032 WRITE(7,5032)JCCLM,JROW,ANSMAT(I,L)
 5032 FORMAT(4X,'C',I4,5X,'R',I1,8X,F11.5)
      GO TO 185
 4034 WRITE(7,5034)JCCLM,JRCW,ANSMAT(I,L)
 5034 FORMAT(4X,'C',I4,5X,'R',I2,7X,F11.5)
      GO TO 185
 4036 WRITE(7,5036)JCOLM,JROW,ANSMAT(I,L)
 5036 FORMAT(4X,'C',I4,5X,'R',I3,6X,F11.5)
      GO TO 185
 4038 WRITE(7,5038)JCOLM,JROW,ANSMAT(I,L)
 5038 FORMAT(4X,'C',I4,5X,'R',I4,5X,F11.5)
  185 CCNTINUE
  190 CCNTINUE
  200 CCNTINUE


C        CARDS FOR THE RIGHT HAND SIDE OF THE CONSTRAINTS

      WRITE(7,6000)
 6000 FORMAT('RHS')
      DO 300 IRHS=1,M
      JRHS=IRHS+N
      IF(JRHS.GT.9) GO TC 7020
      WRITE(7,6010)JRHS,UPPERX(IRHS)
 6010 FORMAT(4X,'B',9X,'R',I1,8X,F11.5)
      GO TO 300
 7020 WRITE(7,6020)JRHS,UPPERX(IRHS)
 6020 FORMAT(4X,'B',9X,'R',I2,7X,F11.5)
  300 CONTINUE
      DC 350 LRHS=1,NN
      KRHS=KK+LRHS
      IF(KRHS.GT.99) GC TC 7C50
      IF(KRHS.GT.9) GO TO 7040
      WRITE(7,6030)KRHS
 6030 FORMAT(4X,'B',9X,'R',I1,13X,'1.0')
      GO TO 350
 7040 WRITE(7,6040)KRHS
 6040 FORMAT(4X,'B',9X,'R',I2,12X,'1.0')
      GO TO 350
 7050 WRITE(7,6050)KRHS
 6050 FORMAT(4X,'B',9X,'R',I3,11X,'1.0')
  350 CCNTINUE
      DO 400 MRHS=1,N
      NRHS=MRHS+KKNN
      IF(NRHS.GT.99) GO TO 7080
      IF(NRHS.GT.9) GO TO 7070
      WRITE(7,6010)NRHS,WPNTGT(MRHS)
      GO TO 400
 7070 WRITE(7,6020)NRHS,WPNTGT(MRHS)
      GC TO 400
 7080 WRITE(7,6080)NRHS,WPNTGT(MRHS)
 6080 FORMAT(4X,'B',9X,'R',I3,6X,F11.5)
  400 CCNTINUE
      STOP
      END
```

LIST OF REFERENCES

1. U.S. National Defense Research Committee Operations Evaluation Group Report No. 56, Theoretical Basis For Methods of Search and Screening, by B. O. Koopman, v. 2B, p. 45-46, 1946.

2. Danskin, J. M., The Theory of Max-Min, p. 85-106, Springer-Verlan, 1967.

3. denBroeder, G. G., Ellison, R. E., and Emerling, L., "On Optimum Target Assignments," Operations Research, v. 7, p. 322-326, May-June 1959.

4. Mylander, C. W., "Applied Mathematical Programming," Proceedings of the United States Army Operations Research Symposium, 4th, 1965, part 1, p. 99-110, March-April 1965.

5. Lemus, F. and David, K. H., "An Optimum Allocation of Different Weapons to a Target Complex," Operations Research, v. 10, p.787-794, September-October 1963.

6. Hadley, G., Nonlinear and Dynamic Programming, p. 104-147 and p. 190-202, Addison-Wesley, 1964.

7. Hillier, F. and Lieberman, G. J., Introduction to Operations Research, p. 574-594, Holden-Day, 1970.

8. Passy, U., "Nonlinear Assignment Problems Treated By Geometric Programming," Operations Research, v. 19, p. 1675-1690, November-December 1971.

9. Manne, A. S., "A Target-Assignment Problem,", Operations Research, v. 6, p. 346-351, May-June 1958.

10. Amerault, J. F., A Computerized Algorithm, M. S. Thesis, U.S. Naval Postgraduate School, Monterey, Californa, 1972.

11. Miercort, F. A. and Soland, R. M., "Optimal Allocation of Missiles Against Area and Point Defenses," Operations Research, v. 19, p. 605-617, May-June 1971.

12. Kooharian, A., Saber, N., and Young, H., "A Force Effectiveness Model With Area Defense," Operations Research, v. 17, p. 895-906, September-October 1969.

13. Perkins, F. M., "Optimum Weapon Deployment For Nuclear Attack," Operations Research, v. 19, p. 77-94, January-February 1961.

14. Day, R. H., "Allocating Weapons To Target Complexes By Means of Nonlinear Programming," Operations Research, v. 14, p. 992-1013, November-December 1966.

15. Eckler, A. R. and Burr, S. A., Mathematical Models of Target Coverage and Missile Allocation, Military Operations Research Society, 1972.

16. Dobbie, J. M., "On the Allocation of Effort Among Deterrent Systems," Operations Research, v. 7, p. 335-346, May-June 1959.

17. Brodheim, E., Herzer, I., and Russ, L. M., "General Dynamic Model For Air Defense," Operations Research, v. 15, p. 779-796, September-October 1967.

18. Kentron Hawaii, Ltd., Anti-Ballistic Missile Allocation To Defend Targets With Time Varying Value Structures, by G. E. Swinson.

19. Charnes, A. and Lemke, C. E., "Minimization of Nonlinear Separable Convex Functions," Naval Research Logistics Quarterly, v. 1, p. 301-312, December 1954.

20. Alloin, G. "A Simplex Method for a Class of Nonconvex Separable Problems," Management Science, v. 17, p. 66-77, September 1970.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Chief of Naval Personnel Pers 116 Department of the Navy Washington, D.C. 20370	1
4.	Naval Postgraduate School Department of Operations Research and Administrative Sciences Monterey, California 93940	1
5.	Assoc. Professor J. G. Taylor, Code 55Tw Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6.	Assoc. Professor W. M. Raike, Code 55Rj Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
7.	Captain Thomas R. McLaughlin Jr USA 124 Euclid Avenue Hamburg, New York 14075	1
8.	Dr. Wilbur B. Payne Deputy Under Secretary of the Army Operations Research 2E621, Pentagon Washington, D.C. 20310	1
9.	Col. H. J. Childress Jr., Commander U.S. Army Combat Development Command Systems Analysis Group Building T-498 Fort Belvoir, Virginia 22060	1

		No. Copies
10.	Commander ARADCOM ATTN: ADGPP-A Ent AFB, Colorado 80912	1
11.	Commanding Officer U.S. Army Small Arms Systems Agency ATTN: Major L. Capps Aberdeen Proving Grounds, Maryland 21005	1

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

REPORT TITLE

A MODIFIED SEPARABLE PROGRAMMING APPROACH TO WEAPON SYSTEM ALLOCATION PROBLEMS

DESCRIPTIVE NOTES (Type of report and, inclusive dates)
Master's Thesis; March 1973

AUTHOR(S) (First name, middle initial, last name)

THOMAS R. McLAUGHLIN JR

REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
March 1973	64	20

a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

This thesis considers mathematical techniques for computing the optimal allocation of weapons from m different systems against n undefended targets. A standard nonlinear programming problem is considered. A discussion is given on John Danskin's Algorithm for the determination of the optimal values of the lagrange multipliers for this problem. Using a transformation of variables, the nonlinear problem is reformulated as a separable problem and solved by separable programming. A new method, the hybrid algorithm, for the determination of the optimal lagrange multipliers is developed.

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6811

UNCLASSIFIED
Security Classification

A-31408

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Weapon System Allocation Problems						
Separable Programming						
Danskin's Algorithm						
Nonlinear Allocation Models						
Computational Algorithm						

Thesis 143248
M245 McLaughlin
c.1 A modified separable
programming approach to
weapon system allocation.